

String manipulation routines

Here is a set of routines to handle null-terminated character strings. Included is a small demonstration as to how they should be used. You will find routines to count the length of a string, copy a whole string, copy a string up to a predetermined offset, concatenate two strings and print a string. Take care to notice which pointer each routine utilizes.

The zero page locations used fall within the floating point accumulators of the BASIC interpreter and should be safe to use, while not performing floating point math. The majority of the strcat routine is adjusting the source pointer to match the length of the existing destination string. This is due to the fact that the second index register cannot be used for indirect indexing.

The routines have been thoroughly tested.

```
; various string manipulation routines by FMan/Tropyx

; all strings handled by these routines must be null-terminated, and their
; maximum length determined by the range of the index registers is 255 chars

!to "str.prg",cbm ; compile using ACME

src = $62
dst = $64
tmp = $66
len = $68

*=$2000

; sample code for using these routines

lda #<str1 ; set address of dst string
sta dst
lda #>str1
sta dst+1
lda #<str2 ; set src string
sta src
lda #>str2
sta src+1
jsr strcat
jsr print ; uses the same dst pointer
rts

str1 !pet "hello, ",0
!fill 8 ; insert work space
str2 !pet "world",13,0

; strlen - returns the length of a string in Y, preserves X
; (upon return, index register Y points to the terminator)
```

```
strlen  ldy #0
slena   lda (dst),y
        beq slenb
        iny
        bne slena
slenb   rts

; strcpy - copies a string from src to dst, preserves X

strcpy  ldy #0
scpya   lda (src),y
        sta (dst),y      ; the store instruction does not change the
        beq scpyb      ; Z flag and this copies the terminator too
        iny
        bne scpya      ; maximum length of the string is 255 chars
scpyb   rts

; strncpy - copies a string up to the specified point set in 'len'

strncpy ldy #0
sncpya  lda (src),y
        sta (dst),y
        beq sncpyb
        iny
        cpy len
        bcc sncpya
        lda #0          ; terminate the destination when cutting
        sta (dst),y
sncpyb  rts

; strcat - concatenates src and dst, ie. adds dst to src - preserves nothing

strcat  jsr strlen      ; get index to the end of the dst string
        lda src+1
        sta tmp+1
        sec
        lda src
        sty tmp          ; subtract the length of the existing
        sbc tmp          ; string from the source address, so that
        sta tmp          ; the index will match
        bcs scata
        dec tmp+1
scata   lda (tmp),y      ; copy source to the end of existing dst
        sta (dst),y
        beq scatb
        iny
        bne scata
scatb   rts

; print - outputs a string to screen (or to a channel specified using
CHKOUT)
```

```
print    ldy #0
pra lda (dst),y
    beq prb
    jsr $ffd2    ; call CHROUT
    iny
    bne pra
prb rts
```

From:

<https://codebase64.org/> - **Codebase 64 wiki**

Permanent link:

https://codebase64.org/doku.php?id=base:string_manipulation_routines

Last update: **2015-04-17 04:34**

